Monitoring for Malware

Ivan Kirillov

18 September 2012

MITRE

© 2012 The MITRE Corporation. All rights reserved.

Agenda

- Hunting Malware Through Artifacts
- OVAL Introduction
- MAEC Introduction
- OVAL & MAEC

© 2012 The MITRE Corporation. All rights reserved.

Malware Artifacts

- When malware executes on a system, it leaves behind ‘artifacts’ that can signify its presence
  - Files
  - Registry Keys
  - Mutual Exclusion (Mutex) Objects
  - Etc.

- Knowledge of these artifacts can be used for detection
  - Especially relevant for 0-day/APT malware
  - Can be used for a malware family, malware instance, or just generic detection

© 2012 The MITRE Corporation. All rights reserved.

Malware Artifact Hunting

- What is needed is some standard way to describe the actions malware performs on a system and the artifacts associated with them..

- How about some standard way to check for these artifacts on an endpoint?

© 2012 The MITRE Corporation. All rights reserved.

What Is OVAL?

- OVAL is a language used to standardize the representation of specific system configurations.
  - XML based
  - Used to describe vulnerable configurations, compliance with specific policy, presence of a patch, etc.
  - Encodes the actual values of registry keys, file attributes, etc. that make up the system state being tested for
  - Removes the ambiguity of textual descriptions

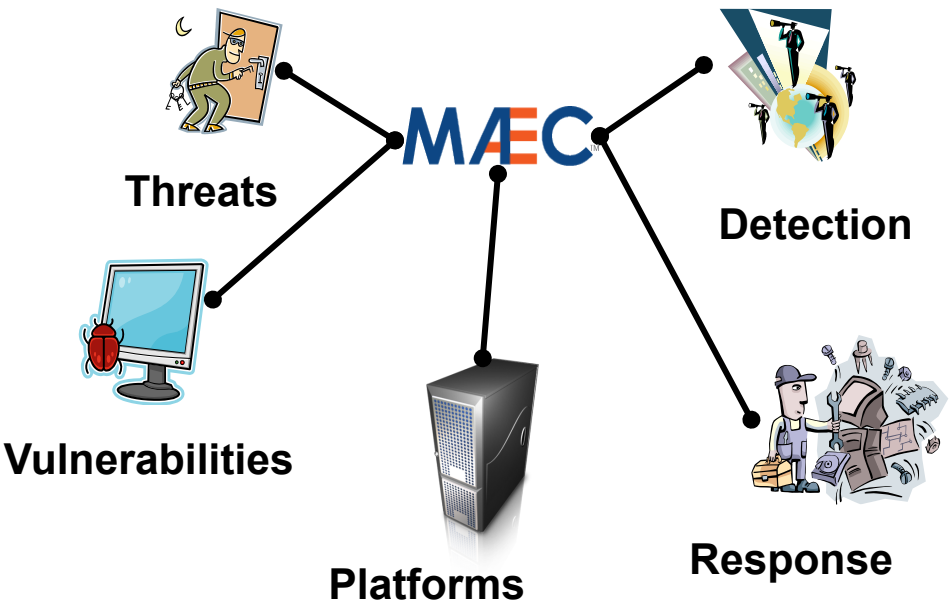
© 2012 The MITRE Corporation. All rights reserved.

Currently Supported Artifacts

- As of OVAL 5.10.1 (for Windows):
  - Files
  - Directories
  - Registry Keys
  - Processes
  - PE Header Entries
  - DNS Cache Entries
  - Ports
  - Services
  - User Accounts

© 2012 The MITRE Corporation. All rights reserved.

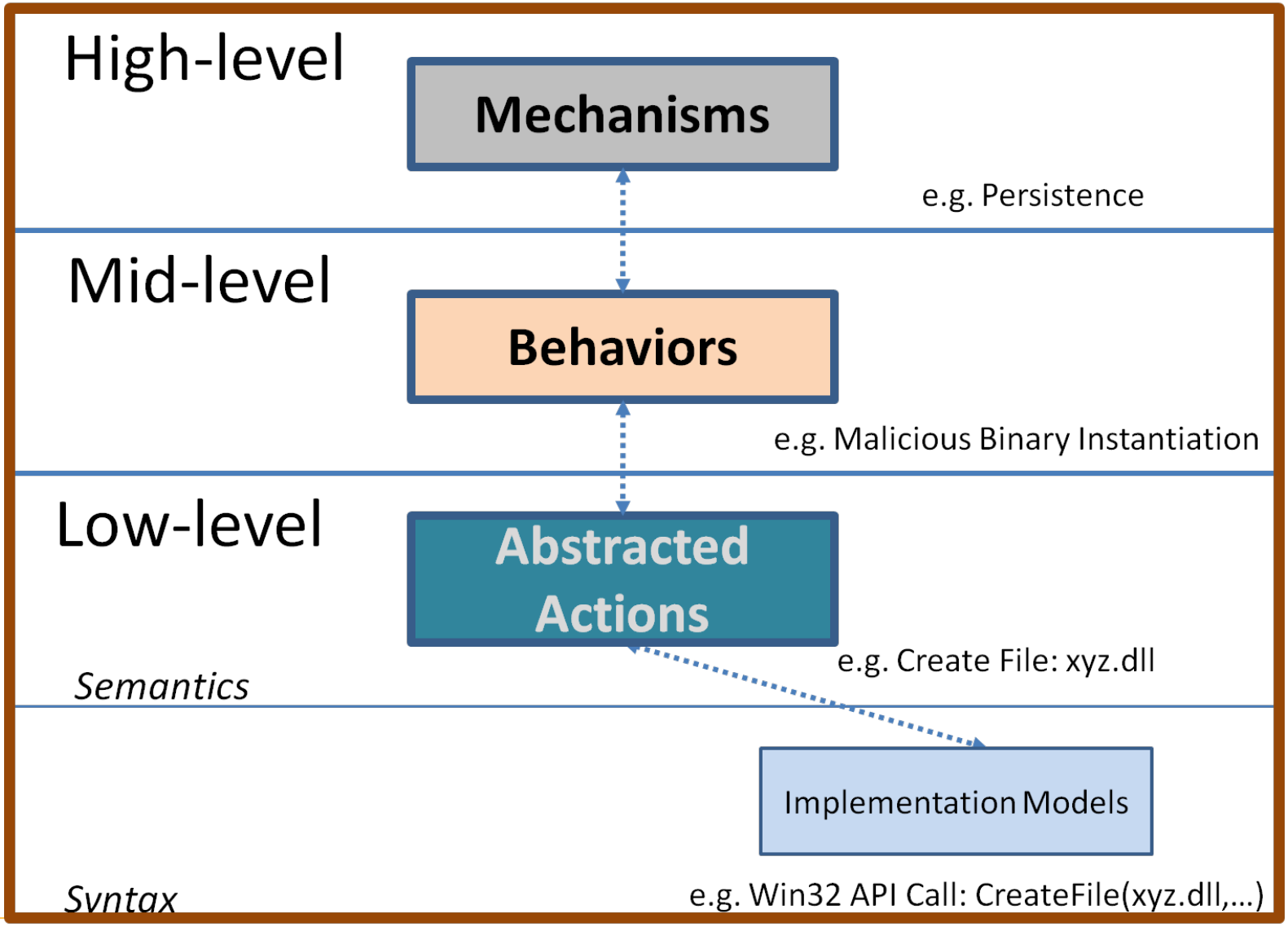
Malware Attribute Enumeration and Characterization (MAEC)



- Language for sharing structured information about malware
  - Grammar (Schema)
  - Vocabulary (Enumerations)
  - Collection Format (Bundle)
- Focus on attributes and behaviors
- Enable correlation, integration, and automation

© 2012 The MITRE Corporation. All rights reserved.

MAEC Structure Overview



© 2012 The MITRE Corporation. All rights reserved.

Bridging the Gap

- MAEC: Provides the intelligence about the malware, including information about associated artifacts and how they were utilized/affected
- OVAL: Provides the checking language and means for assessing a system for malware artifacts
- MAEC + OVAL: Together, these two standards allow for comprehensive and adaptable artifact-based malware detection
  - With the help of scripts, the entire process can be automated
  - OVAL on endpoints = enterprise level detection
  - Standards basis means information on the malware and the detection can be easily shared/consumed

© 2012 The MITRE Corporation. All rights reserved.

Tools: MAEC → OVAL Translator Script

- Python-based
- Converts MAEC v2.1 XML into OVAL v5.7 XML
  - Looks for actions that created files or registry key
- Additional object support forthcoming
- Available on MAEC Github:
  - https://github.com/MAECProject/Tools

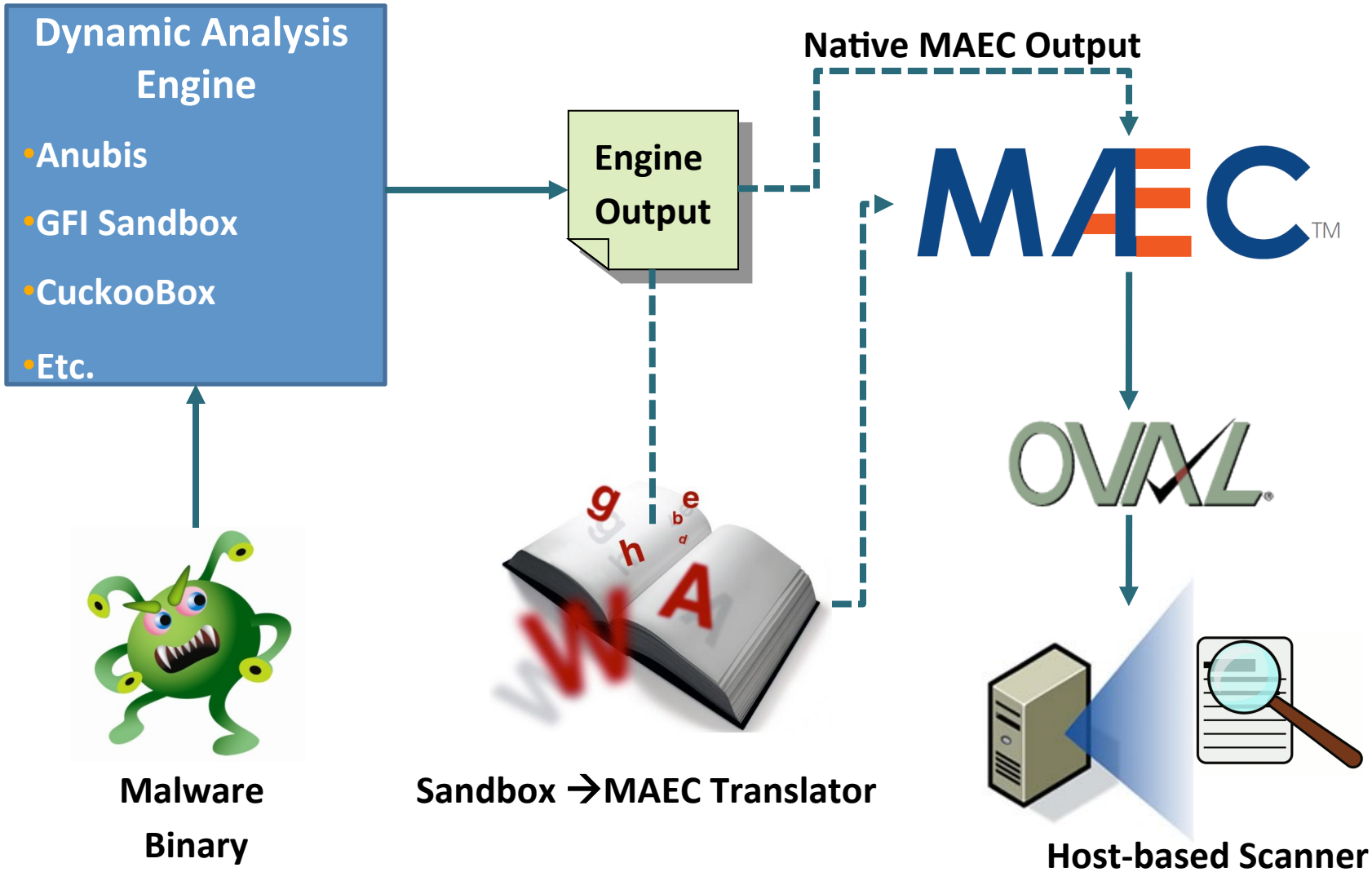
© 2012 The MITRE Corporation. All rights reserved.

Tools: OVAL Interpreter

- Freely available reference implementation
- Demonstrates usability of the OVAL Language
- Drives the development of the OVAL Language
- Validate & test content
- A reference for developers
- Reduce the cost of OVAL adoption
- Available on OVAL website:
  - http://oval.mitre.org/language/interpreter.html

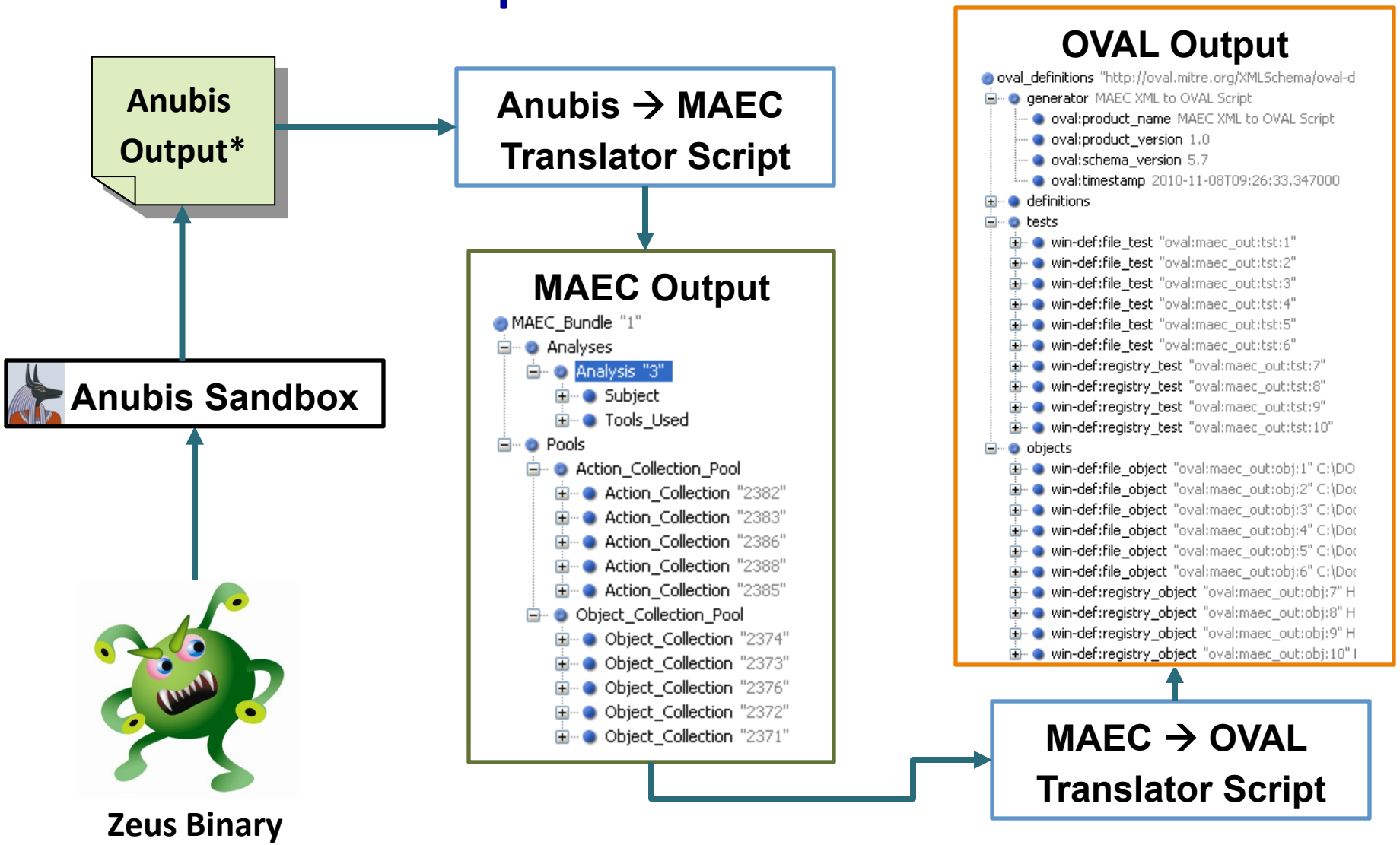
© 2012 The MITRE Corporation. All rights reserved.

MAEC & OVAL Interaction



© 2012 The MITRE Corporation. All rights reserved.

Real World Example: MAEC & Zeus Bot



*http://anubis.iseclab.org/?action=result&task_id=1167a57d1aa905e949df5d5478ab23bf9

© 2012 The MITRE Corporation. All rights reserved.

Questions?

http://maec.mitre.org

http://oval.mitre.org

© 2012 The MITRE Corporation. All rights reserved.